

Оптимальное распределение вычислительных ресурсов для реализации сервисов цифровых валют центральных банков

А.С. Албычев, Д.Ю. Ильин, Е.В. Никульчев

Аннотация—Рассматривается реализация цифровой валюты центрального банка (ЦВЦБ) как распределенного вычислительного сервиса с взаимосвязанными компонентами в условиях максимальной загруженности. В условиях, когда общая производительность сервиса зависит от максимальной пропускной способности одного из компонентов, ставится задача перераспределения ресурсов между компонентами сервиса. В качестве целевой функции оптимального решения введено отклонение от максимального количества имеющихся ресурсов суммы зависимостей затрат процессорных ресурсов каждого компонента от объема обрабатываемых запросов. Задача оптимизации состоит в поиске разбиения ресурсов между задействованными компонентами сервиса при минимизации целевой функции. Разработан метод оптимального распределения ресурсов на основе сформулированной задачи и идентификации зависимостей затрат процессорных ресурсов компонентами на основе проведения нагрузочных испытаний на масштабируемом экспериментальном стенде для шести компонентов ЦВЦБ. При построении моделей учитывалась специфика обработки данных при операциях с ЦВЦБ. Полученные результаты демонстрирует эффективность разработанных моделей и метода поиска оптимального разбиения.

Ключевые слова—вычислительные ресурсы, виртуальные машины, производительность высоконагруженных систем, оптимальное распределение ресурсов, цифровая валюта центрального банка.

I. ВВЕДЕНИЕ

Цифровая валюта центрального банка (ЦВЦБ) с точки зрения программно-аппаратной реализации представляет собой распределенный вычислительный сервис [1], обеспечивающий ведение учета по перемещению виртуальных денежных средств, эквивалентных национальной денежной единице. В настоящее время исследователями во многих странах ведутся финансовые, экономико-математические, информационно-вычислительные исследования по реализации, внедрению и эффективному

использованию ЦВЦБ [2-5]. ЦВЦБ реализуются как с использованием блокчейн-технологий [6], так и без них. Одной из возможностей цифровой валюты является хранение всех транзакций по всем единицам хранения [7], что определяет высоконагруженность вычислительных сервисов, предоставляющих информационное обеспечение.

Для высоконагруженных вычислительных сервисов стоит задача эффективного использования вычислительных ресурсов, то есть оправданное выделение ресурсов под конкретные модули и компоненты вычислительных сервисов. Загруженность компонентов сервиса зависит, с одной стороны, от объема и вида поступающих данных, при этом выделяемые ресурсы могут быть определены экспериментальным образом с использованием виртуальной среды [8]. С другой стороны, когда сервис представляет собой взаимодействующие распределенные компоненты, пропускная способность каждого компонента зависит не только от исходных запросов, но и от производительности взаимосвязанных компонентов, так называемого «узкого горлышка», имеющего место в любом вычислительном комплексе. Например, компонент 1 использует все выделенные вычислительные ресурсы, в то время как другие, использующие те же данные, даже при имеющихся свободных ресурсах не могут выполнить больше команд, чем получено от компонента 1. Таким образом, необходимо перераспределить ресурсы между взаимосвязанными компонентами так, чтобы в высоконагруженной работе объем незадействованных ресурсов был минимальным.

В статье разработан метод оптимального распределения ресурсов центрального процессора (ЦП) между взаимосвязанными виртуальными машинами, реализующими программно-аппаратные компоненты ЦВЦБ. Построена информационная модель на виртуальных машинах, реализующая ЦВЦБ, математическая модель зависимости загруженности процессоров от объема обрабатываемых запросов, сформулирована задача оптимизации, разработана структура экспериментальных исследований для идентификации параметров математической модели, проведено экспериментальное исследование и сравнение с теоретическими результатами найденного оптимального распределения ресурсов. Новизна полученных результатов заключается в идентифицируе-

Статья получена 27 апреля 2024.

А.С. Албычев, Федеральное казначейство Министерства финансов Российской Федерации, МИРЭА — Российский технологический университет, Москва, Россия (e-mail: albychev@mirea.ru).

Д.Ю. Ильин, МИРЭА — Российский технологический университет, Москва, Россия (e-mail: i@dmitryilin.com).

Е.В. Никульчев, МИРЭА — Российский технологический университет, Москва, Россия (e-mail: nikulchev@mail.ru).

мых моделях, задаче оптимизации вычислительных ресурсов, предназначенных для высоконагруженных вычислительных сервисов ЦВЦБ, отличающихся от известных введенной целевой функцией, представляющей собой комбинацию ограничений, зависящих от производительности компонента, использующего все доступные ресурсы.

II. ПРОГРАММНО-АППАРАТНАЯ МОДЕЛЬ ЦВЦБ

Реализован экспериментальный стенд, имитирующий вычислительный сервис обеспечения ЦВЦБ [9-11], состоящий из 7 виртуальных машин, 6 из которых выполняли роль компонентов ЦВЦБ и 1 – роль генератора нагрузки (рис. 1).

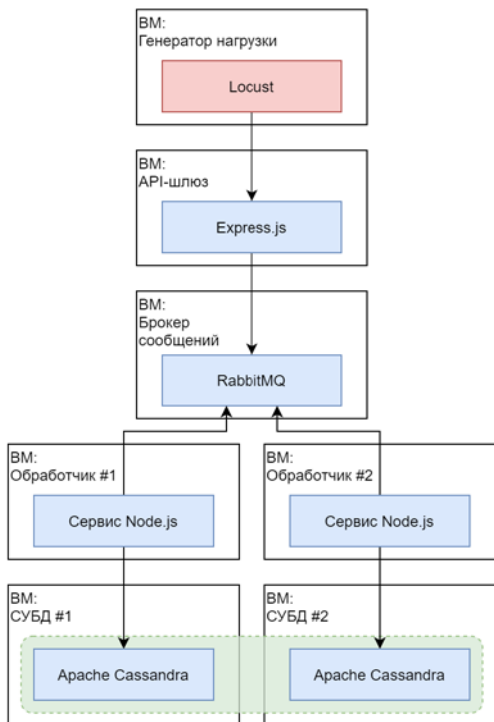


Рис. 1. Схема структуры организации экспериментального стенда и исполняемого программного обеспечения модели сервиса ЦВЦБ

В представленной схеме «Генератор нагрузки» создает виртуальных пользователей. Виртуальные пользователи выполняют HTTP-запросы к «API-шлюзу», который, в свою очередь, отправляет их в «Брокер сообщений» и помещает в очередь, из которой «Обработчик #1» и «Обработчик #2» извлекают сообщения для обработки. «API-шлюз» при этом ожидает ответного сообщения в очереди, прежде чем вернуть ответ по протоколу HTTP. «Обработчик» при получении сообщения из очереди выполняет операцию и отправляет в соответствующую «СУБД» запросы на изменение и вставку записей. После получения ответа от «СУБД» «Обработчик» помещает в очередь сообщений ответ об успешном выполнении операции. «API-шлюз» получает сообщение и возвращает ответ виртуальному пользователю генератора нагрузки.

Экспериментальный стенд был размещен на следующем аппаратном обеспечении:

- процессор: AMD Ryzen 7 3700X 3.6ГГц;

- ОЗУ: 32 ГБ;

- дисковая система: Samsung SSD QVO 2TB.

Программное обеспечение экспериментального стенда:

- операционная система хост-машины: Windows 10 Pro;

- гипервизор: VirtualBox версии 6.1.30;

- гостевая операционная система: Ubuntu 20.04.6 LTS.

Версии программного обеспечения, использованного при запуске модели ЦВЦБ:

- Apache Cassandra версии 4.1.3;

- Locust версии 2.21.0;

- RabbitMQ версии 3.8.2;

- Node.js версии 18.17.1;

- Express.js версии 4.18.2.

Для очереди сообщений используется протокол AMQP 0-9-1 по push-модели. Для запросов в Apache Cassandra задан уровень согласованности по умолчанию (уровень One, т. е. ответ на запрос должен предоставить хотя бы один узел). Пространство ключей в СУБД Apache Cassandra было настроено с коэффициентом репликации 2, т. е. каждая запись реплицировалась на обе виртуальные машины.

III. МОДЕЛЬ ОПТИМИЗАЦИИ И МЕТОД РЕШЕНИЯ

Пусть задано шесть ВМ в соответствии со схемой на рис. 1, при генерации запросов x зап/сек количество используемых процессорных ресурсов будет функционально зависеть от x :

$$y_i = f_i(x), i = \overline{1,6}, \quad (1)$$

где $f_i(x)$ – гладкие, монотонно возрастающие функции, независимые друг от друга, которые могут быть идентифицированы на основе экспериментальных исследований.

Задано общее количество ресурсов K .

Необходимо в условиях (1) для каждого y_i найти ограничение k_i , такое, что

$$y_i \leq k_i, i = \overline{1,6}, \sum_{i=1}^6 k_i = K; \quad (2)$$

$$J = K - \sum_{i=1}^6 y_i \rightarrow 0. \quad (3)$$

Иными словами, нужно так перераспределить ресурсы между шестью компонентами, обеспечивающими сервис ЦВЦБ, чтобы объем незадействованных ресурсов был минимальным, при условии, что можно определить (например, на основании экспериментов) зависимость затрат ресурсов ЦП от объема обрабатываемых данных.

Разработан метод решения задачи (1)-(3), состоящий из следующих шагов.

1. Реализация экспериментального стенда, обеспечивающего масштабируемый сервис ЦВЦБ в условиях заданных ограничений на общий процессорный ресурс K .
2. Формирование алгоритма экспериментального исследования с нагрузкой, соответствующей реаль-

ным запросам к сервису.

3. Генерация входного потока запросов с нарастающей нагрузкой до предельной в условиях некоторого разбиения K на k_i ($i = \overline{1, 6}$).
4. Обработка данных, полученных на основе проведенных нагрузочных испытаний, и идентификация моделей (1).
5. Поиск решения задачи (1)–(3) в условиях найденных зависимостей $f_i(x)$.
6. Экспериментальная проверка теоретического решения: проведение нагрузочных испытаний с полученным разбиением ресурсов k_i , вычисление экспериментального значения целевой функции (3).

IV. МАСШТАБИРУЕМЫЕ ЭКСПЕРИМЕНТАЛЬНЫЕ ИССЛЕДОВАНИЯ

В соответствии со схемой, приведенной на рис. 1, реализован экспериментальный стенд. Для обеспечения масштабируемости общий объем выделенных процессорных ресурсов принят $K=800$.

Нагрузочные испытания, сбор и обработка экспериментальных данных производились по единому алгоритму, представленному на рис. 2. Используемые конфигурации ВМ, настройки пространства ключей и конфигурации нагрузки идентичны, за исключением количества ядер ЦП и ограничений по времени ядра ЦП.

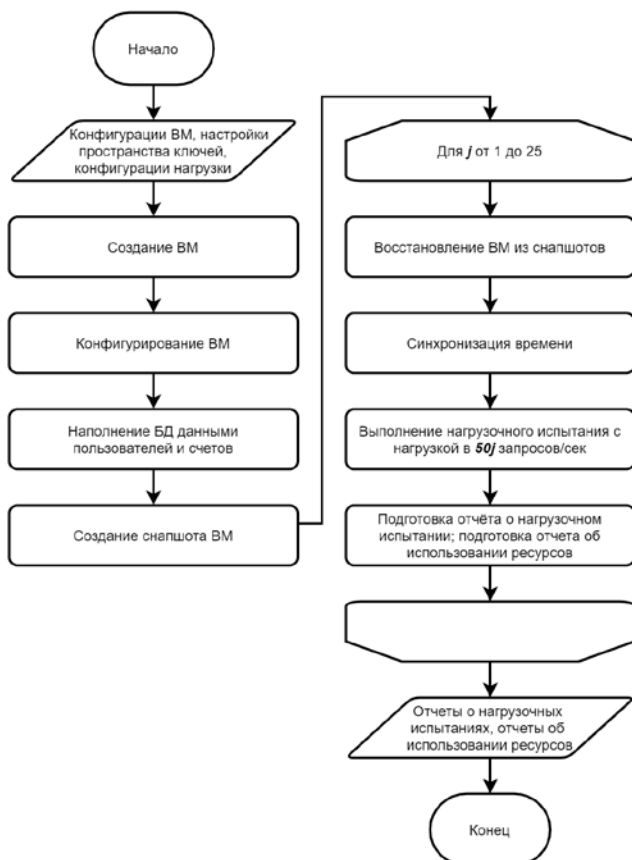


Рис. 2. Общий алгоритм проведения экспериментов

На подготовительном этапе производилось создание виртуальных машин и их настройка, включая соответствующее программное обеспечение. Затем база данных

наполнялась 1 млн записей о пользователях и 1 млн записей о счетах. После заполнения создавались снимоты (снимки) виртуальных машин, готовых к проведению нагрузочных испытаний.

Проводилась серия из $n=25$ нагрузочных испытаний. Виртуальные машины восстанавливались из снимота, после чего проводилась синхронизация времени и формирование целевой нагрузки l_j , где $j = \overline{1, 25}$.

В каждом нагрузочном испытании виртуальные пользователи Locust были сконфигурированы для выполнения нагрузки, не превышающей 2.5 запроса в секунду. Количество виртуальных пользователей на каждом шаге было равно $20j$, а прирост числа пользователей в секунду, до достижения заданного уровня целевой нагрузки, был равен $2j$. Таким образом, выполнялись нагрузочные испытания с целевой нагрузкой $l_j = 50j$, т. е. нагрузкой, увеличивающейся на каждом шаге на 50 запросов в секунду.

По итогу каждого нагрузочного испытания формировался отчет Locust, а также экспортировались отдельные отчеты об использовании ресурсов ЦП по каждой виртуальной машине.

По каждому нагрузочному испытанию для каждой виртуальной машины формируется отчет, содержащий уровень использования ресурсов ЦП за каждую секунду эксперимента.

При обработке данных отчетов об использовании вычислительных ресурсов выполнялись следующие шаги:

1. Пропуски заполнялись значениями за предшествующую секунду.
2. Выполнялось обнаружение участка в 600 секунд, во время которого проводилась нагрузка, остальные замеры отбрасывались.
3. Выбросы (отрицательные значения и значения, превышающие выделенное процессорное время, умноженное на количество ядер) заменялись на значения, соответствующие границам.
4. Производилось вычисление математического ожидания для данных об использовании ЦП.

Извлеченные данные о математическом ожидании по каждой виртуальной машине объединялись с отчетом о проведенном нагрузочном испытании для последующего анализа.

V. ИДЕНТИФИКАЦИЯ МОДЕЛЕЙ

Для идентификации моделей проведены экспериментальные исследования с произвольным распределением ресурсов между шестью компонентами k_i . Виртуальным машинам были выделены ресурсы, представленные в табл. 1. Для ВМ не задавались ограничения по времени ядра, а на каждый слой архитектуры ЦВЦБ было выделено по 2 ядра ЦП.

После проведения серии нагрузочных испытаний были получены отчеты о пропускной способности модели сервиса ЦВЦБ. В табл. 2 показано, как изменяется фактическая пропускная способность по мере увеличения нагрузки. С заданного уровня нагрузки в 800 запросов в

секунду наблюдается стагнация в отношении повышения числа обрабатываемых запросов.

ТАБЛИЦА 1 – ЗАДАНИЕ ПАРАМЕТРЫ ВИРТУАЛЬНЫХ МАШИН

ВМ	ЦП, кол. ядер	Время ядра, тики	ОЗУ, ГБ
Генератор нагрузки	4	100	2
API-шлюз	2	100	1
Брокер сообщений	2	100	1
Обработчик #1	1	100	1
Обработчик #2	1	100	1
СУБД #1	1	100	2
СУБД #2	1	100	2

Как видно из табл. 2, максимальный уровень числа обрабатываемых запросов не превышает 781.5163 запроса в секунду. По мере увеличения нагрузки вырастет также среднее время ответа, что согласуется с настройками виртуальных пользователей Locust.

ТАБЛИЦА 2 – ОЦЕНКИ ПРОПУСКНОЙ СПОСОБНОСТИ ПРИ ИНФРАСТРУКТУРЕ С РАСПРЕДЕЛЕНИЕМ РЕСУРСОВ НА ОСНОВЕ ЭКСПЕРТНОЙ ОЦЕНКИ

Нагрузка, запросы/сек	Пропускная способность, запросы/сек	Среднее время ответа, мс	Средний объем свободных ресурсов ЦП, тики
50	48.9051	42.6043	686.4758
100	98.8906	39.9333	612.7913
150	148.1972	59.1269	575.4825
200	198.2722	72.2949	531.6661
250	247.9629	58.7410	449.0985
300	296.8385	92.6427	460.0634
350	347.3128	97.2142	401.4007
400	395.1900	122.1140	383.8948
450	446.3347	131.2374	383.9199
500	495.5722	187.7537	333.1636
550	538.6977	245.3518	325.2988
600	584.8866	293.5404	313.1035
650	618.4665	341.9627	318.1386
700	650.8521	375.7756	314.9316
750	686.7718	391.2822	308.6244
800	702.3511	426.1200	305.4608
850	725.1637	443.6361	296.0067
900	748.6596	460.2285	299.8197
950	741.6275	497.8828	300.5392
1000	772.2201	504.2026	296.8865
1050	757.0539	544.1691	298.8247
1100	767.0819	564.1533	299.7629
1150	743.3609	610.2197	298.7930
1200	781.5163	605.7285	296.4407
1250	778.3480	634.0605	304.0968

Отобразив данные о среднем использовании ресурсов

ЦП (рис. 3) в зависимости от целевой нагрузки, можно заметить, что обе виртуальные машины компонентов «СУБД» с Apache Cassandra используют все выделенные ресурсы. На рис. 3 видно, что, начиная с уровня в 400 запросов в секунду, задействуется около 100% времени ЦП.

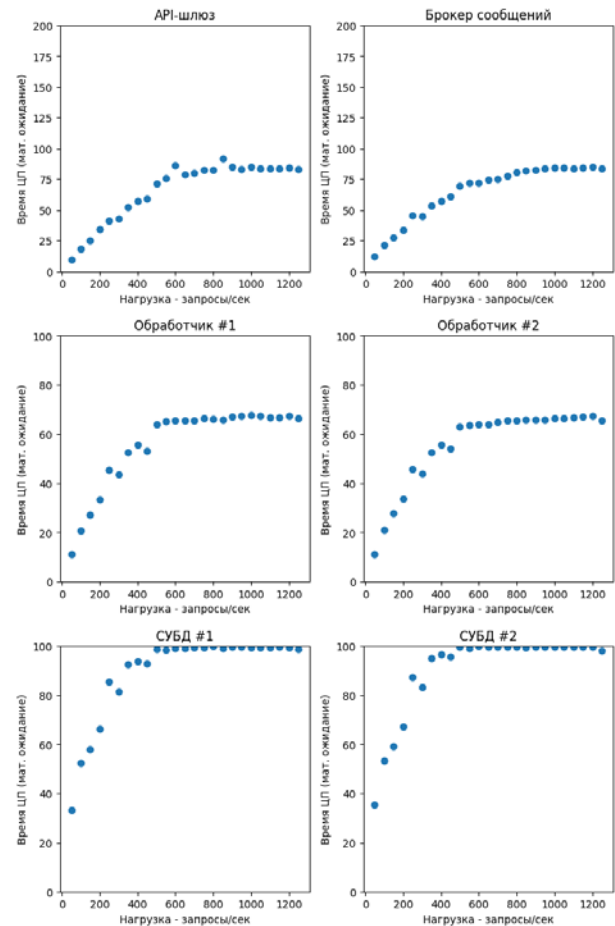


Рис. 3. Показатели среднего задействования времени ЦП в зависимости от целевой нагрузки

На данном эксперименте, проводимом для идентификации модели, видна особенность рассматриваемой задачи. Компоненты «СУБД #1», «СУБД #2» достигают максимума выделенного ресурса в точке, соответствующей нагрузке около 400 запросов в секунду. При этом же значении нагрузки достигают предельного значения использования процессорных ресурсов остальные взаимосвязанные компоненты. Таким образом, например, увеличение ресурсов для «Брокера сообщений» никак не повлияет на общую производительность, поскольку «узким местом» здесь являются ресурсы, выделенные для компонентов «СУБД». Полученные данные подтверждают корректность задачи (1)-(3).

Учитывая характер полученных данных, можно принять функции в (1) в линейном виде, при этом для идентификации используются данные до своих предельных значений (до пологих участков соответствующих графиков). Используя данные о задействованных ресурсах ЦП до найденного уровня нагрузки в 400 запросов в секунду, были построены линейные модели (рис. 4). Синими точками обозначены замеры, полученные до достижения

использования полного объема вычислительных ресурсов одной из виртуальных машин, а красными – после достижения. Зеленой линией показаны линейные модели, а пунктирной линией отмечен объем выделенных ресурсов ЦП.

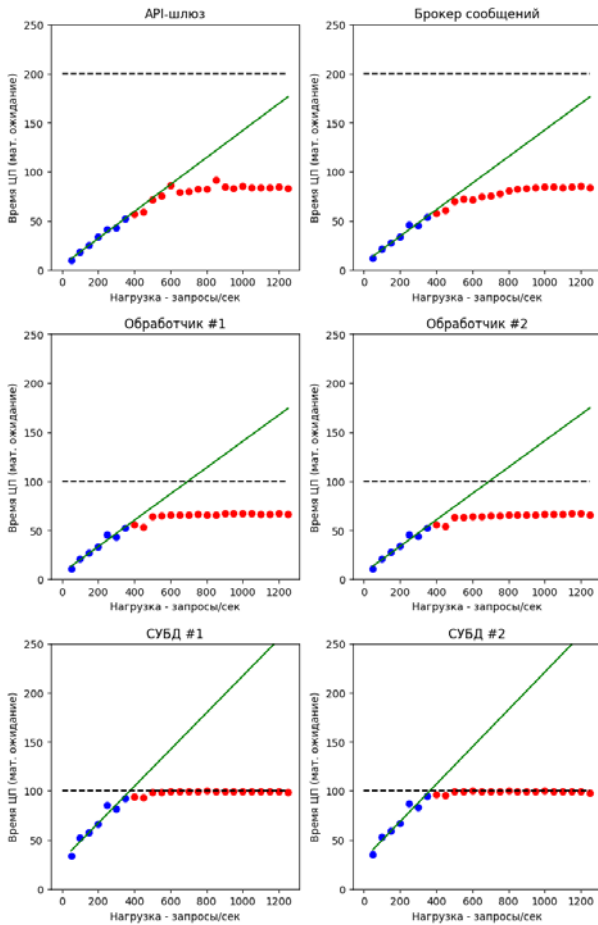


Рис. 4. Линейные модели затрат ресурсов ЦП в зависимости от целевой нагрузки

Обозначим потребляемую нагрузку каждого компонента: y_1 — «API-шлюз»; y_2 — «Брокер сообщений»; y_3 — «Обработчик #1»; y_4 — «Обработчик #2»; y_5 — «СУБД #1»; y_6 — «СУБД #2», x — нагрузка. Тогда система уравнения (1) с параметрами, идентифицируемыми на основе эксперимента, будет иметь вид:

$$\begin{aligned}
 y_1 &= 0.1376x + 4.4307; \\
 y_2 &= 0.1354x + 7.1164; \\
 y_3 &= 0.1342x + 6.6008; \\
 y_4 &= 0.1343x + 6.8784; \\
 y_5 &= 0.1877x + 29.4992; \\
 y_6 &= 0.1902x + 30.6113.
 \end{aligned}
 \tag{4}$$

VI. ОПТИМАЛЬНОЕ РАСПРЕДЕЛЕНИЕ РЕСУРСОВ

В условиях найденных моделей (4) решим задачу (1)–(3), т. е. необходимо в условиях (4) для каждого y_i найти ограничение k_i , такое, что

$$y_i \leq k_i, \quad i = \overline{1, 6}, \quad \sum_{i=1}^6 k_i = 800;$$

$$J = 800 - \sum_{i=1}^6 y_i \rightarrow 0$$

Решение данной задачи очевидно. Из уравнения $J = 0$, подставляя уравнения из системы (4), находим критическое значение $x_{кр}$:

$$x_{кр} = 777.5898.$$

Подставляя значения в (4) и приравнявая $k_i = y_i$, получаем искомое распределение ресурсов:

$$\begin{aligned}
 k_1 &= 111.4128; \\
 k_2 &= 112.3664; \\
 k_3 &= 110.9337; \\
 k_4 &= 111.3365; \\
 k_5 &= 175.4164; \\
 k_6 &= 178.5341.
 \end{aligned}$$

Таким образом, теоретическое решение найдено. Для оценки достоверности проведем сравнение теоретического результата с экспериментальными значениями.

В соответствии с полученными оценками было проведено перераспределение ресурсов ЦП (табл. 3). Виртуальным машинам были установлены ядра ЦП и ограничения на время ядра, соответствующие оценкам на основе моделей.

ТАБЛИЦА 3 – ЗАДАННЫЕ И ВЫЧИСЛЕННЫЕ ПАРАМЕТРЫ ВИРТУАЛЬНЫХ МАШИН

ВМ	ЦП, кол. ядер	Время ядра, тики	ОЗУ, ГБ
Генератор нагрузки	4	100	2
API-шлюз	2	56	1
Брокер сообщений	2	56	1
Обработчик #1	2	56	1
Обработчик #2	2	56	1
СУБД #1	2	88	2
СУБД #2	2	88	2

В случае превышения уровня в 1 ядро и 100 тиков времени виртуальным машинам (ВМ) назначалось 2 ядра, для которых задавалось ограничение по времени. Значения также были скорректированы, исходя из требований целочисленного типа данных. Значения параметров, отличающиеся от табл. 2, выделены в табл. 3 полужирным курсивом.

На экспериментальном стенде с перераспределенными ресурсами ЦП была проведена серия нагрузочных испытаний. На рис. 5 и в табл. 4 приведен уровень фактической производительности по мере увеличения нагрузки. Заметная стагнация в отношении числа обрабатываемых запросов наблюдается после нагрузки в 1100 запросов в секунду.

Из табл. 4 видно, что максимальный уровень числа обрабатываемых запросов моделью сервиса ЦВЦБ достигает 954.2838 запроса в секунду. По мере увеличения нагрузки, как и в первом эксперименте, вырастает среднее время ответа. Однако при высокой нагрузке оно ни-

же, чем при произвольном распределении ресурсов.

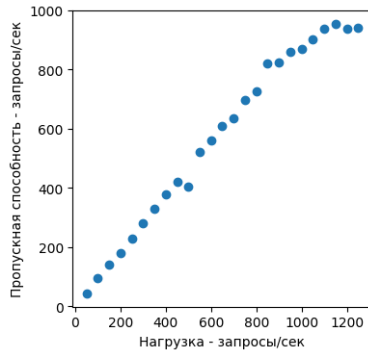


Рис. 5. Показатели пропускной способности модели сервиса ЦВЦБ в зависимости от целевой нагрузки после перераспределения ресурсов

ТАБЛИЦА 4 – Оценки пропускной способности для ИНФРАСТРУКТУРЫ С РАСПРЕДЕЛЕНИЕМ РЕСУРСОВ НА ОСНОВЕ ПРЕДЛОЖЕННОГО МЕТОДА

Нагрузка, запросы/сек	Пропускная способность, запросы/сек	Среднее время ответа, мс	Средний объем свободных ресурсов ЦП, тики
50	44.5947	80.3336	625.3823
100	95.8191	55.7040	498.8047
150	142.6736	66.7153	459.1553
200	180.5712	104.2168	447.6678
250	230.4296	100.8436	357.0985
300	282.3206	116.5228	382.0050
350	330.2252	127.4006	337.4591
400	377.1159	128.9022	258.7780
450	421.4804	146.0458	236.8214
500	403.5352	228.8908	292.7713
550	519.8181	177.2931	160.4925
600	561.9751	187.0312	175.2788
650	608.3979	205.0485	150.8915
700	636.0103	233.9468	163.4007
750	697.7407	260.1756	106.6694
800	724.9086	287.0579	129.5342
850	820.8133	282.3659	78.2204
900	824.8766	304.6717	111.2504
950	859.3860	334.1597	110.3372
1000	870.2509	365.6972	114.9599
1050	900.5932	385.3228	108.6795
1100	936.5267	409.8609	102.7496
1150	954.2838	422.1929	84.8564
1200	938.4528	474.1304	131.6711
1250	939.8610	500.5904	135.0634

На основе данных о среднем использовании ресурсов ЦП (рис. 6) можно заметить сокращение объема задействованных ресурсов. Обе виртуальные машины компонентов «СУБД» с Apache Cassandra достигают использования всех выделенных ресурсов, но уже начиная с уровня нагрузки в 850 запросов в секунду. Значения на рис. 6, превышающие заданные ограничения по выделенным ресурсам, приведены к заданной границе.

При сравнении результатов нагрузочных испытаний

можно отметить, что при максимальной нагрузке в 1250 запросов в секунду получен прирост в числе обрабатываемых запросов на 20.75%. При этом среднее время обработки запроса при той же нагрузке сократилось на 21.05%, а объем задействованных ресурсов сократился на 55.59%.

Продолжение роста производительности, наблюдаемое при достижении полного задействования ресурсов процессора, может быть связано с особенностями ЦП, использованного в хост-машине.

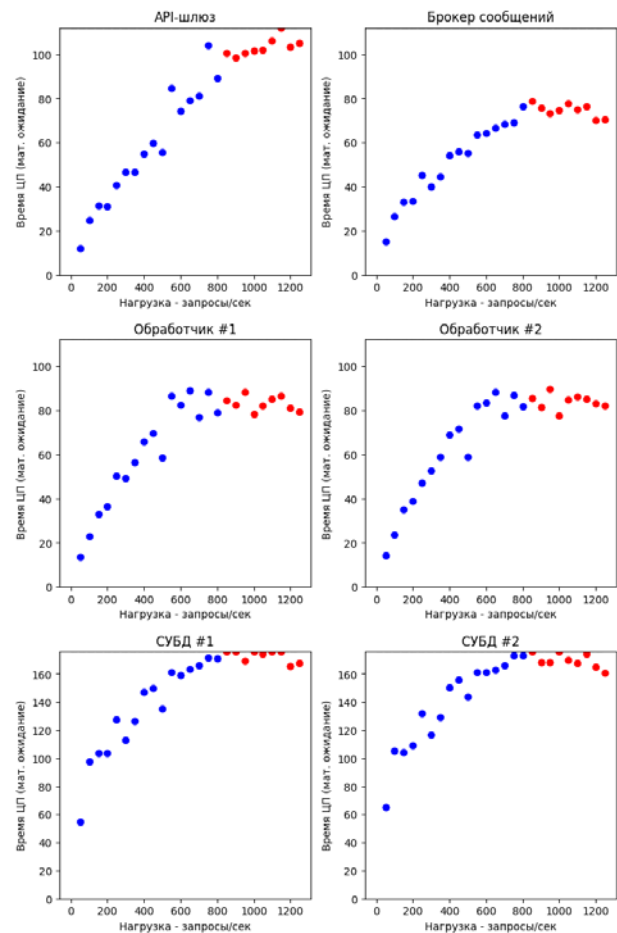


Рис. 6. Показатели среднего задействования времени ЦП в зависимости от целевой нагрузки после перераспределения ресурсов

Отмеченный выход среднего значения задействованных ресурсов ЦП за установленные ограничения может быть связан с неточностями в соблюдении ограничений на уровне гипервизора или с неточностями в работе системы мониторинга, осуществляющей сбор данных об использовании вычислительных ресурсов.

VII. ЗАКЛЮЧЕНИЕ

В статье рассматривается реализация ЦВЦБ как распределенного вычислительно сервиса с взаимосвязанными компонентами. Для оптимального распределения ресурсов в условиях максимальной загруженности предложены модели и метод, основанный на идентификации по экспериментальным нагрузочным испытаниям. Предложена целевая функция оптимального размещения, представляющая собой минимизацию отклонения от максимального количества имеющихся ресурсов суммы зави-

симостей затрат ресурсов ЦП каждым компонентом, реализующим сервис ЦВЦБ, от объема обрабатываемых запросов. Разработан масштабируемый экспериментальный стенд, который позволяет рассмотреть отличное от представленного количество виртуальных машин и компонентов, реализующих модель сервиса ЦВЦБ. Предложенный метод оптимального распределения ресурсов на основе проведения нагрузочных испытаний на масштабируемом экспериментальном стенде позволил повысить количество обрабатываемых запросов в секунду на 20%.

БИБЛИОГРАФИЯ

- [1] Албычев А.С., Кудж С.А. Среда исследований операционно-вычислительной архитектуры информационного обеспечения цифровой валюты центрального банка // *Russian Technological Journal*. 2023. Т. 11, № 3. С. 7-16.
- [2] Allen S., Ćapkun S., Eyal I., et al. Design choices for central bank digital currency: Policy and technical considerations. National Bureau of Economic Research (No. w27634). 2020.
- [3] Auer R., Böhme R. The technology of retail central bank digital currency. *BIS Quarterly Review*, 2020.
- [4] Alora A., Sahoo P., Sasidharan A. Central bank digital currency adoption challenges-The case of an emerging nation // *Finance Research Letters*. 2024. Vol. 59, P. 104756.
- [5] Hoang Y.H., Ngo V.M., Vu N.B. Central bank digital currency: A systematic literature review using text mining approach // *Research in International Business and Finance*. 2023. Vol. 64. P. 101889.
- [6] Garriga M., Dalla Palma S., Arias M., De Renzis A., Pareschi R., Tamburri, D.A. Blockchain and cryptocurrencies: A classification and comparison of architecture drivers // *Concurrency and Computation: Practice and Experience*. 2021. Vol. 33, No. 8. P. e5992.
- [7] Genc H.O., Takagi S. A literature review on the design and implementation of central bank digital currencies // *International Journal of Economic Policy Studies*. 2024. Vol. 18. P. 197–225.
- [8] Ильин Д.Ю. Программный фреймворк для экспериментальной оценки характеристик информационно-технологических решений в виртуальной среде // *Russian Technological Journal*. 2022. Т. 10, № 5. С. 16-27.
- [9] Албычев А.С., Ильин Д.Ю. Выбор стека технологий вычислительной инфраструктуры для экспериментальных исследований цифровых валют // *International Journal of Open Information Technologies*. 2023. Т. 11, № 4. С. 95-101.
- [10] Ильин Д.Ю., Никульчев Е.В., Албычев А.С. Разработка программно-конфигурируемых вычислительных стендов для исследований современных информационных технологий // В сб.: *Инженерное образование в цифровом обществе. Материалы Международной научно-методической конференции*. Ч. 1.— Минск, 2024. С. 251-253.
- [11] Албычев А.С. Исследование программно-аппаратной модели цифровой валюты центрального банка с историей транзакций // *Вестник современных цифровых технологий*. 2024. № 18. С. 13-23.

Optimal computational resource allocation for implementation of central bank digital currency services

A.S. Albychev, D.Y. Ilin, E.V. Nikulchev

Abstract—The implementation of the central bank digital currency (CBDC) as a distributed computing service with interconnected components is considered under conditions of high load. When the overall performance of the service depends on the maximum throughput of one of its components the necessity to redistribute resources between the service components arises. The deviation of the sum of CPU resource utilization of each service component from the given resource volume dependent on given workload is introduced as an objective function for finding the optimal solution. The optimization goal is to find the allocation of computational resources between the involved service components while minimizing the objective function. A method for optimal resource allocation has been developed, which is based on the described optimization problem and identification of functional relation of CPU resource utilization of each service component to the given workload, that is achieved with load testing using scalable experimental setup consisting of six CBDC service components. The specifics of data processing during operations with CBDC were considered in the process of the models' construction. The results obtained demonstrate the effectiveness of the developed models and the method for finding the optimal resource allocation.

Keywords—computational resources, virtual machines, high load system performance, optimal resource allocation, central bank digital currency.

REFERENCES

- [1] A.S. Albychev & S.A. Kudzh, "Development of a research environment for the operational and computational architecture of central bank digital currency software," *Russian Technological Journal*, vol. 11, no. 3, pp. 7-16, 2023.
- [2] S. Allen, S. Čapkun, I. Eyal, ... & F. Zhang, *Design choices for central bank digital currency: Policy and technical considerations*, (No. w27634). National Bureau of Economic Research.
- [3] R. Auer & R. Böhme, *The technology of retail central bank digital currency*. BIS Quarterly Review, March, 2020.
- [4] A. Alora, P. Sahoo & A. Sasidharan, "Central bank digital currency adoption challenges-The case of an emerging nation," *Finance Research Letters*, vol. 59, p. 104756, 2024.
- [5] Y.H. Hoang, V.M. Ngo & N.B. Vu, "Central bank digital currency: A systematic literature review using text mining approach," *Research in International Business and Finance*, vol. 64, p. 101889, 2023.
- [6] M. Garriga, S. Dalla Palma, M. Arias, A. De Renzis, R. Pareschi & D.A. Tamburri, "Blockchain and cryptocurrencies: A classification and comparison of architecture drivers," *Concurrency and Computation: Practice and Experience*, vol. 33, no. 8, p. e5992, 2021.
- [7] H.O. Genc & S. Takagi, "A literature review on the design and implementation of central bank digital currencies," *International Journal of Economic Policy Studies*, vol. 18, pp. 197–225, 2024.
- [8] D. Ilin, "Framework for experimental evaluation of software solutions in a virtual environment," *Russian Technological Journal*, vol. 10, no. 5, pp. 16-27, 2022.
- [9] A.S. Albychev & D.Y. Ilin, "Vybor steka tehnologij vychislitel'noj infrastruktury dlja jeksperimental'nyh issledovaniy cifrovyyh valjut," *International Journal of Open Information Technologies*, vol. 11, no. 4, pp. 95-101, 2023. [In Rus]
- [10] D.Y. Ilin, E.V. Nikulchev, A.S. Albychev, "Razrabotka programmno-konfiguriruemykh vychislitel'nykh stendov dlja issledovaniy sovremennykh informacionnykh tehnologij," In *Inzhenernoe obrazovanie v cifrovom obshhestve*. Materialy Mezhdunarodnoj nauchno-metodicheskoy konferencii. Ch. 1. Minsk, 2024, pp. 251-253. [In Rus]
- [11] A.S. Albychev, "Issledovanie programmno-apparatnoj modeli cifrovoj valjuty central'nogo banka s istoriej tranzakcij," *Vestnik sovremennykh cifrovyykh tehnologij*, vol. 18, pp. 13-23, 2024. [In Rus]